

İçindekiler

1. Arduino Programının Yüklmesi.....	2
2. İki Buton ile Bir LED'i Yakıp Söndüren Uygulama	3
3. PinMode Uygulaması.....	5
4. Yürüyen ışık uygulaması	8
5. Nokia sms sesi uygulaması	11
6. PWM ile LED Parlaklığının Kontrolü	13
7. Trafik Lambası Uygulaması	16
8. Arduino ile LCD Ekran (16x2) Yazı Yazma – Kayan Yazı Uygulaması.....	18
9. Arduino NTC Uygulaması.....	21
10. İleri – Geri Step Motor Kontrolü.....	24
11. DA Motor Hız ve Yön Kontrolü	27
12. Arduino ile Servo Motor Kontrolü.....	30



Hayatı Kodluyoruz.



Temrin Adı:

Arduino Programının Yüklmesi

Temrin No:

1

ÖĞRENCİNİN ADI

DEĞERLENDİRME

ADI:

TEKNOLOJİ

İŞLEM BASAMAKLARI

İŞ ALIŞKANLIĞI

SÜRE

TOPLAM

SOYADI:

%30

%30

%30

%10

Rakam
ile

Yazı
ile

Ders Öğretmeni

Yunus ÖZEN

Koordinatör:

Yaşar SEVER



Hayatı Kodluyoruz.



Temrin Adı:

İki Buton ile Bir LED'i Yakıp Söndüren Uygulama

Temrin No: 2

Amaç: Arduino Uno'nun 2 numaralı dijital pinine bağlanan butona basıldığında led yanacak. 3 numaralı dijital pine bağlı butona basıldığında ise led sönecektir. Bu temrinde gerekli yazılım ve bağlantı şemasını öğreneceğiz.

Devrede kullanılan malzemeler:

- 1x Arduino uno
- 1x LED
- 2x Buton
- 1x 10 k Ω Direnç
- 1x 220 Ω Direnç

Aşağıdaki arduino yazılımını yazınız ve arduinoya yükleyiniz.

Gerekli yazılım:

```
const int buton1 = 2;
const int buton2 = 3;
const int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buton1, INPUT);
  pinMode(buton2, INPUT);
}

void loop(){
  if (digitalRead(buton1)==HIGH){
    digitalWrite(ledPin, HIGH); // LED yak
  }
  else if (digitalRead(buton2)==HIGH){
    digitalWrite(ledPin, LOW); // LED söndür
  }
}
```

Break-Board üzerine aşağıdaki devre şemasını kurunuz.



Hayatı Kodluyoruz.



Temrin Adı:

PinMode Uygulaması

Temrin No: 3

Arduinoda **pinMode()** fonksiyonu pinlerin giriş olarak mı yoksa çıkış olarak mı ayarlanması gerektiğini belirtir. Giriş olarak ayarlanan pinlere dış dünyada veri girişi sağlar. Örneğin sensörlerden, butonlardan gelen verileri okumaya ve işlemeye hazır hale getirir. Çıkış olarak ayarlandığında ise dış dünyadaki yükler kontrol edilebilir. Örneğin pine bağlı olan LED yakılıp söndürülebilir, röle, motor, lamba gibi yükler kontrol edilebilir.

pinMode(pin, MODE);

pin parametresi ayarlanacak olan pinin numarası,

mode parametresi ise **INPUT** veya **OUTPUT** olmalı, giriş olarak ayarlanacaksa **INPUT**, çıkış olarak ayarlanacaksa **OUTPUT** yazılmalıdır.

Delay() fonksiyonu ise arduinoda bekleme komutudur. Fonksiyon içine yazılan sayı mili saniye cinsinden bir bekleme sağlar. Örneğin 1 saniyelik bekleme sağlamak için gerekli fonksiyonu **delay(1000)** şekilde olmalıdır.

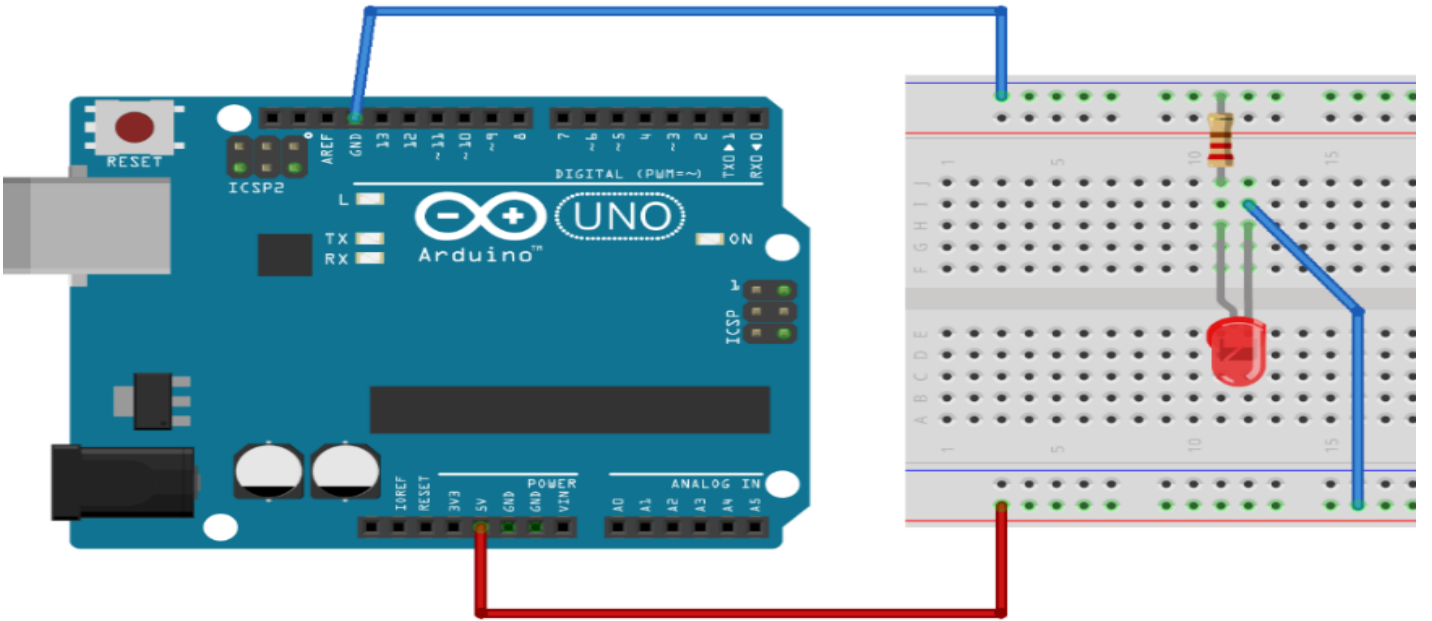
Devrede kullanılan malzemeler:

- 1x Arduino uno
- 1x LED
- 1x Buton
- 1x 10 kΩ Direnç
- 2x 220 Ω Direnç

Şimdide **pinMode()** fonksiyonunu kullanarak arduinonun 13 numaralı pinine bağlı LED'i bir saniye aralıklar ile yanıp sönmelerini sonra ise 2 numaralı pine bağlı buton ile yanıp sönmelerini sağlayalım.

1. Uygulama için gerekli olan yazılımı yazınız ve Arduino Uno'ya yükleyiniz.

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```



Birinci uygulama şeması

2. Uygulama için gerekli olan yazılım.

```
int buton = 2, led = 13;  
  
void setup() {  
  pinMode(buton, INPUT);  
  pinMode(led, OUTPUT);  
}  
  
void loop() {  
  if(digitalRead(buton)==HIGH) {  
    digitalWrite(led,HIGH);  
  }  
  else {  
    digitalWrite(led,LOW);  
  }  
}
```


Temrin Adı:

Yürüyen ışık uygulaması

Temrin No:

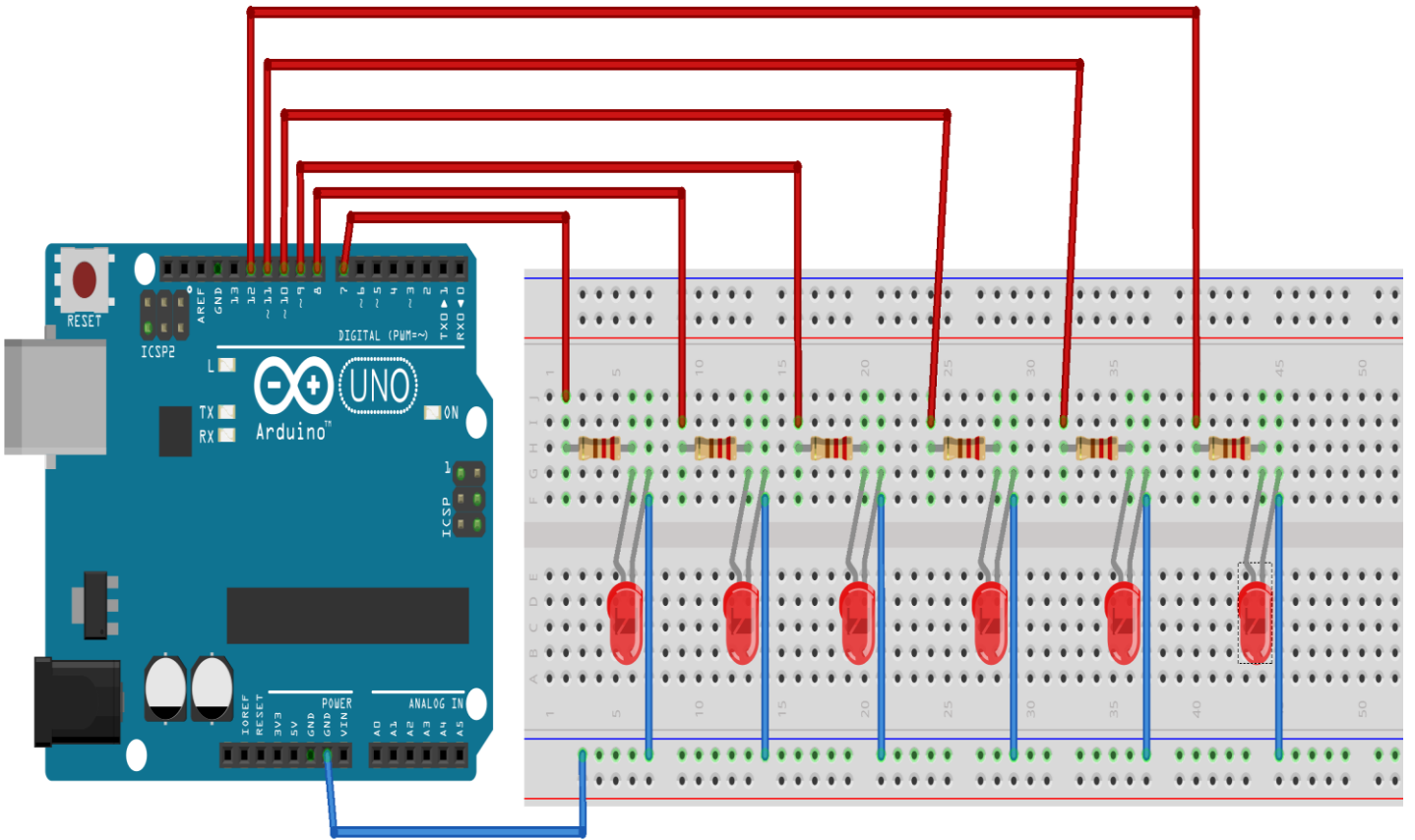
4

Bu uygulamada arduino uno'nun 7, 8, 9, 10, 11, 12 numaralı pinlerine bağlı ledleri sıra ile yakıp söndüreceğiz. Gerekli devre şemasını ve kod yapısını öğreneceğiz.

Devrede kullanılan malzemeler:

- 1x Arduino uno
- 6x LED
- 6x 220 Ω Direnç

Devre şeması;



Arduino kodu1;

```
int led1 = 7, led2 = 8, led3 = 9, led4 = 10, led5 = 11, led6 = 12 ;  
int bekle = 100;
```

```
void setup() {
```

```
    pinMode(led1, OUTPUT);  
    pinMode(led2, OUTPUT);  
    pinMode(led3, OUTPUT);  
    pinMode(led4, OUTPUT);  
    pinMode(led5, OUTPUT);  
    pinMode(led6, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    digitalWrite(led1, HIGH);  
    delay(500);  
    digitalWrite(led1, LOW);  
    delay(500);
```

```
    digitalWrite(led2, HIGH);  
    delay(500);  
    digitalWrite(led2, LOW);  
    delay(500);
```

```
    digitalWrite(led3, HIGH);  
    delay(500);  
    digitalWrite(led3, LOW);  
    delay(500);
```

```
    digitalWrite(led4, HIGH);  
    delay(500);  
    digitalWrite(led4, LOW);  
    delay(500);
```

```
    digitalWrite(led5, HIGH);  
    delay(500);  
    digitalWrite(led5, LOW);  
    delay(500);
```

```
    digitalWrite(led6, HIGH);  
    delay(500);  
    digitalWrite(led6, LOW);  
    delay(500);
```

```
}
```




Hayatı Kodluyoruz.



Temrin Adı:

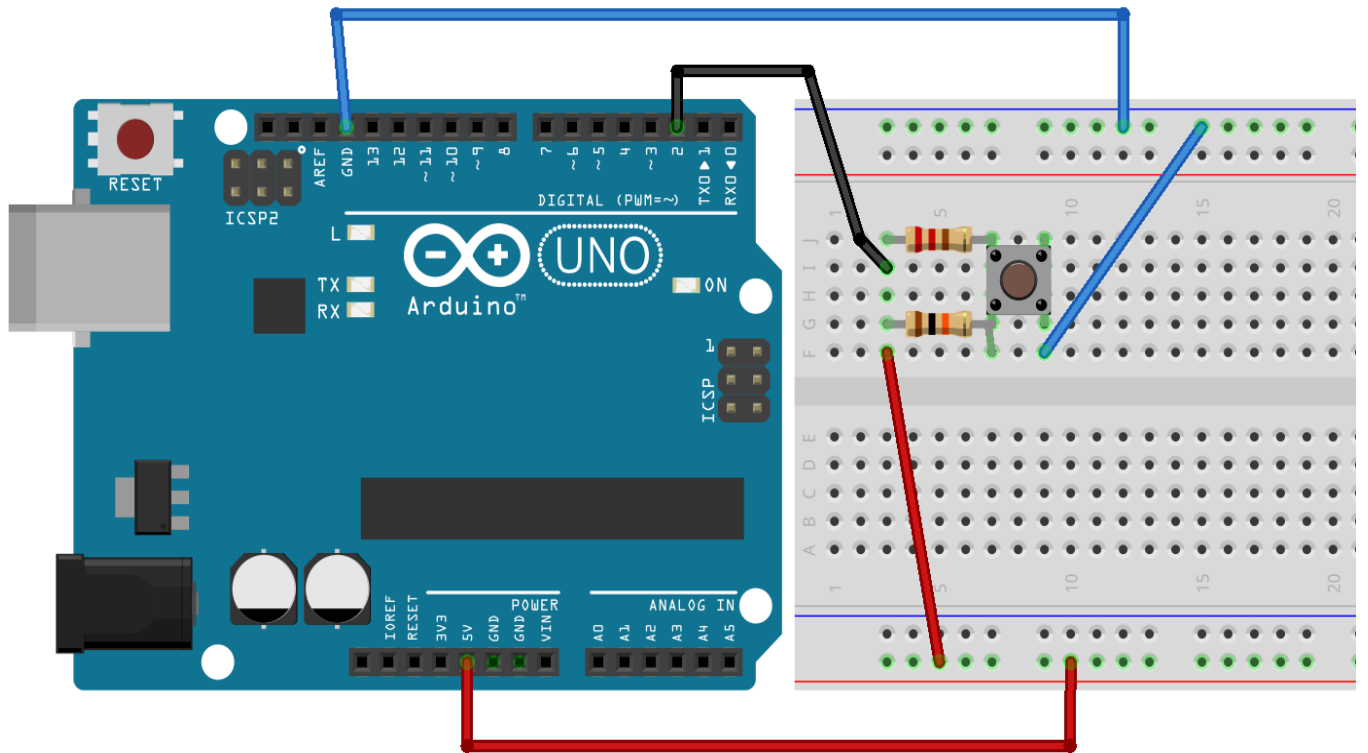
Nokia sms sesi uygulaması

Temrin No:

5

Bu uygulamada arduino uno'nun 2 numaralı pinine bağlı butona basıldığında 13 numaralı pine bağlı hoparlörden klasik Nokia 3310 mesaj sesi veren uygulamayı inceleyeceğiz.

Devrenin bağlantı şeması;



Arduino UNO kodu;

```
int buton=2, ses_pin=13;
void setup() {
  pinMode(buton, INPUT_PULLUP);
  pinMode(ses_pin, OUTPUT);
}
void loop() {
  if(digitalRead(buton)==LOW){
    digitalWrite(ses_pin, HIGH);
    delay(100);
    digitalWrite(ses_pin, LOW);
    delay(100);

    digitalWrite(ses_pin, HIGH);
    delay(100);
    digitalWrite(ses_pin, LOW);
    delay(100);

    digitalWrite(ses_pin, HIGH);
    delay(100);
    digitalWrite(ses_pin, LOW);
    delay(300);

    digitalWrite(ses_pin, HIGH);
    delay(300);
    digitalWrite(ses_pin, LOW);
    delay(200);

    digitalWrite(ses_pin, HIGH);
    delay(300);
    digitalWrite(ses_pin, LOW);
    delay(200);

    digitalWrite(ses_pin, HIGH);
    delay(100);
    digitalWrite(ses_pin, LOW);
    delay(100);
    delay(500);
  }
}
```

ÖĞRENCİNİN ADI		DEĞERLENDİRME				
ADI:	TEKNOLOJİ	İŞLEM BASAMAKLARI	İŞ ALIŞKANLIĞI	SÜRE	TOPLAM	
SOYADI:	% 30	% 30	% 30	% 10	Rakam ile	Yazı ile
	Ders Öğretmeni	Yunus ÖZEN	Koordinatör:	Yaşar SEVER		

Temrin Adı:

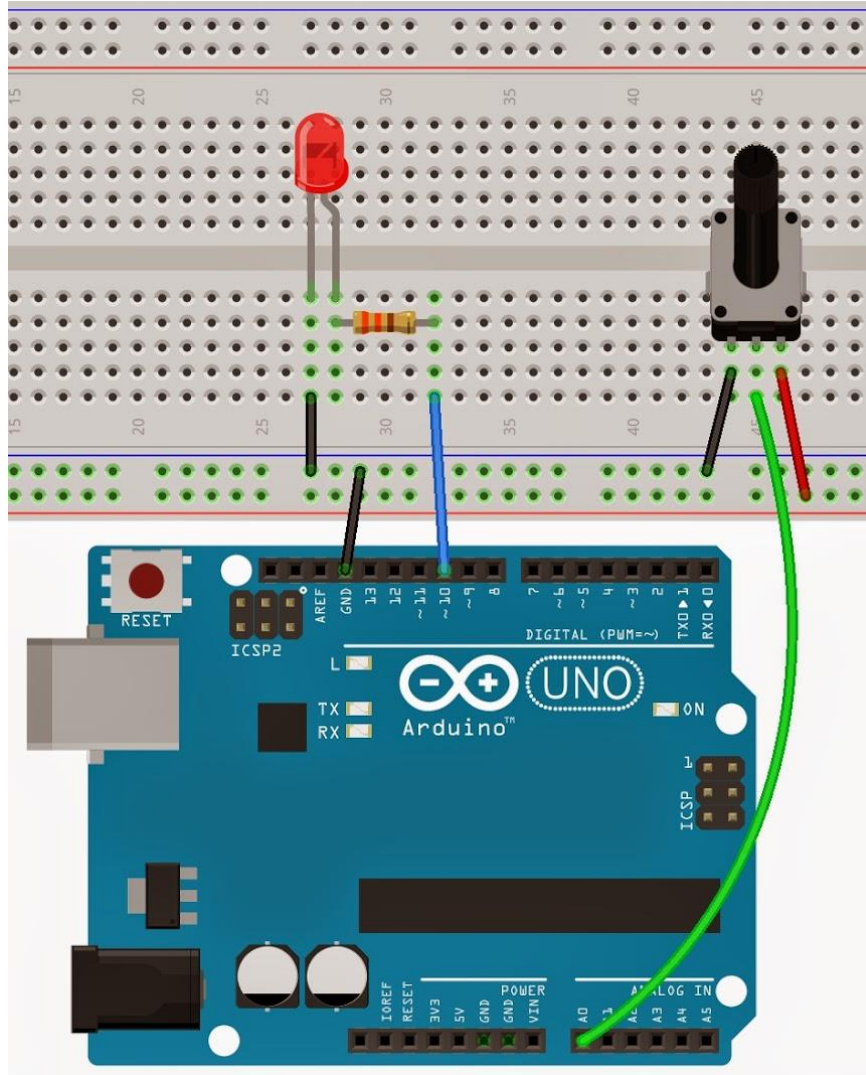
PWM ile LED Parlaklığının Kontrolü

Temrin No: 6

Amaç: Led in bağlanacağı pin 3 nolu pwm pinidir. Loop metodu içerisinde for döngüsüyle i sayacı değeri 0 dan başlatılarak 255 e kadar saydırılıyor. Led en az ışık düzeyinden en yüksek düzeye çıkarak bu işlemi devam ettirir. 10 ms lik gecikme süresini değiştirerek gözlemde bulununuz.

Devrede kullanılan malzemeler:

- 1x Arduino uno
- 1x LED
- 1x 220 Ω Direnç
- 1x10 k Ω Potansiyometre

Devre Şeması:

Arduino Kodu:

```
int led=10;
int pot=A0;
int potDeger;

void setup(){

}

void loop(){
  potDeger=analogRead(pot);
  potDeger=map(potDeger, 0, 1023, 0, 255);
  analogWrite(led, potDeger);
  delay(10);
}
```

Kısa Bilgi:

map() Fonksiyonu:

Bu fonksiyon bir oranlama fonksiyonudur. Belirli bir aralıktaki sayının, başka bir aralıkta hangi sayıya karşılık geldiğini hesaplar. map fonksiyonuna parametre olarak beş değer girilir.

map(sayi, aralik_1_altdeger, aralik_1_ustdeger, aralik_2_altdeger, aralik_2_ustdeger);

Yukarıdaki ifadede sayi, aralik_1_altdeger ile aralik_1_ustdeger arasında bir değerdir. map fonksiyonunun ürettiği yeni değer ise aralik_2_altdeger ile aralik_2_ustdeger arasında bir değerdir.

Not: analogRead fonksiyonu 0-1023 arası bir değer üretir, analogWrite fonksiyonu ise 0-255 arası bir değeri çıkışlardan verir. Biz burada map fonksiyonu ile analogRead'den aldığımız 0-1023 skalasındaki değeri 0-255 sklasında bir değere çevireceğiz ve bu değeri çıkış olarak yazdıracağız.

Notlar:

.....

.....

.....

.....

.....

.....

.....

ÖĞRENCİNİN ADI	DEĞERLENDİRME					
	TEKNOLOJİ	İŞLEM BASAMAKLARI	İŞ ALIŞKANLIĞI	SÜRE	TOPLAM	
	%30	%30	%30	%10	Rakam ile	Yazı ile
	Ders Öğretmeni	Yunus ÖZEN	Koordinatör:	Yaşar SEVER		



Hayatı Kodluyoruz.

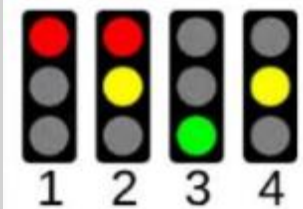


Temrin Adı:

Trafik Lambası Uygulaması

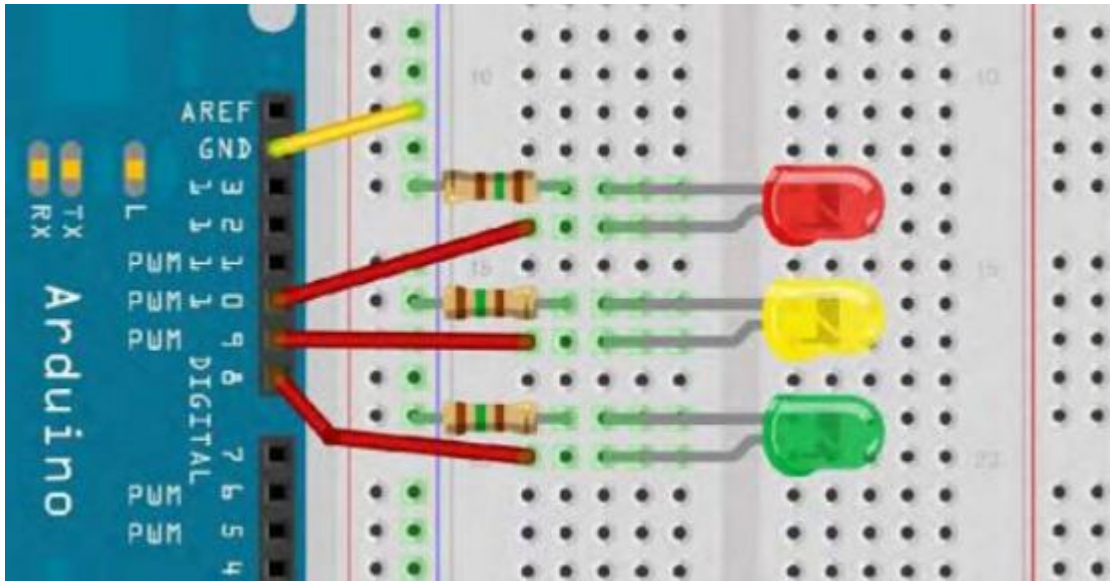
Temrin No: 7

Açıklama: Sarı ledin yanıp sönme süresi 2 saniyedir. Led sürelerini istediğiniz gibi “delay(....)” komutu içerisine yazarak değiştirebilirsiniz. Ledler aşağıdaki sırada çalışır.



Devrede Kullanılan Malzemeler:

- 1x Arduino UNO
- 1x Kırmızı LED
- 1x Sarı LED
- 1x Yeşil LED

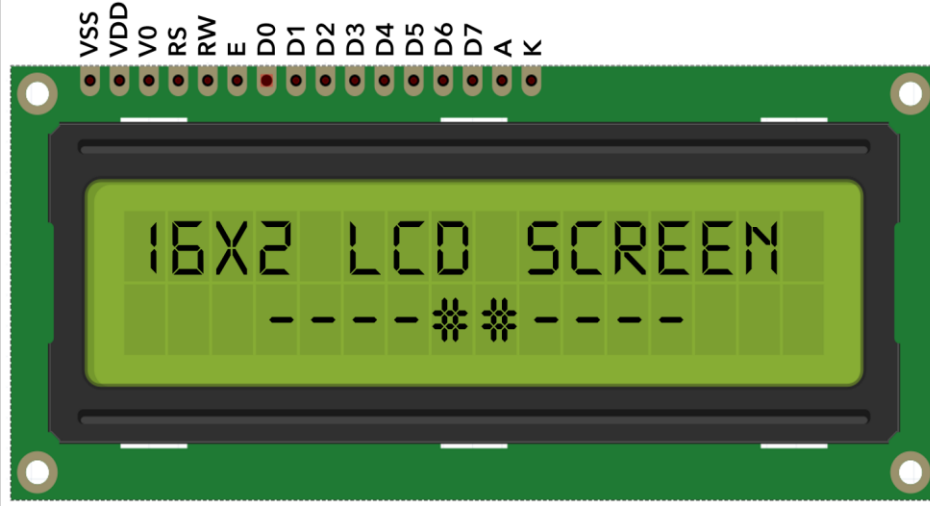


Temrin Adı:

Arduino ile LCD Ekran (16x2) Yazı Yazma – Kayan Yazı Uygulaması

Temrin No:

8

Açıklama:

No	Symbol	Function
1	VSS	Ground
2	VDD	5V +
3	V0	Contrast
4	RS	Register
5	RW	Read/Write
6	E	Enable
7	D0	Data bus
8	D1	Data bus
9	D2	Data bus
10	D3	Data bus
11	D4	Data bus
12	D5	Data bus
13	D6	Data bus
14	D7	Data bus
15	A	Anode (5V+)
16	K	Cathode (GND)

Devrede Kullanılan Malzemeler:

- 1x Arduino UNO
- 1x Kırmızı LED

Notlar:

[illegible]

ÖĞRENCİNİN ADI	DEĞERLENDİRME					
ADI:	TEKNOLOJİ	İŞLEM BASAMAKLARI	İŞ ALIŞKANLIĞI	SÜRE	TOPLAM	
SOYADI:	%30	%30	%30	%10	Rakam ile	Yazı ile
	Ders Öğretmeni	Yunus ÖZEN	Koordinatör:	Yaşar SEVER		

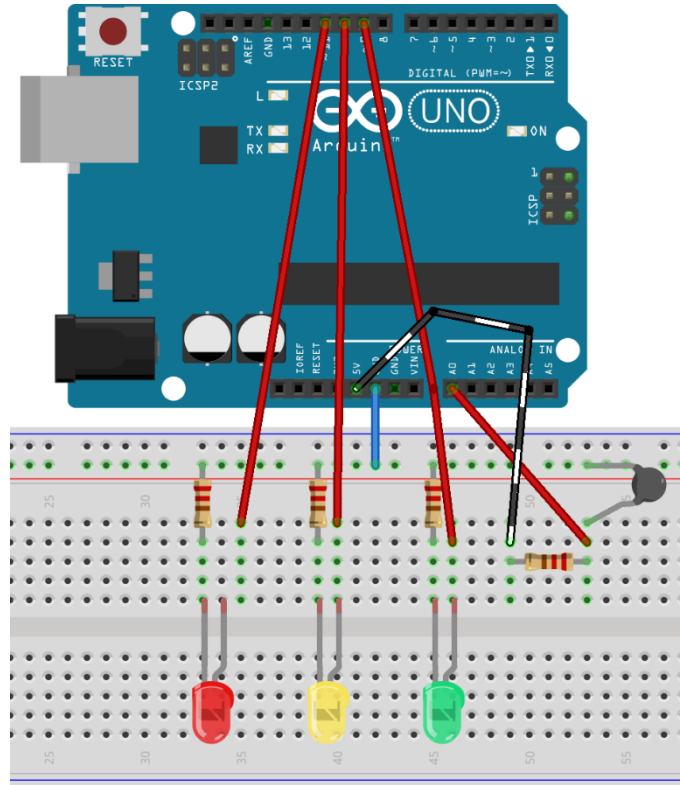
Temrin Adı: Arduino NTC Uygulaması

Temrin No: 8

Devrede Kullanılan Malzemeler:

- 1x Arduino UNO
- 1x NTC
- 3x LED (KIRMIZI, SARI YEŞİL)
- 1X 470 Ω
- 3X 220 Ω

Açıklama: NTC' ler (Negative Temperature Coefficient) sıcaklığı arttıkça direnci azalan devre elemanlarıdır. Ntc' den okumak istediğimiz analog gerilimi alabilmemiz için, değişen direnç değerini gerilim bölücü direnç sayesinde analog gerilime dönüştürmemiz gerekir. Aşağıdaki devremizde ntc ile 560 Ω ' luk direncin birleştiği yerden A0 pinine bağlantı sağlanarak gerilim bölücü elde edilmiştir. Ntc' nin oda sıcaklığındaki direnci 500 Ω civarında olduğundan sıcaklık arttıkça A0 pinine giden gerilim düşecektir. Bu sayede elde ettiğimiz analog sinyalleri yazılımla yorumlayacağız.

Devre Şeması:

ARDUİNO KODU:

```
int sensorSicaklik=0;
int kirmiziLed=9;
int yesilLed=10;
int maviLed=11;
void setup()
{
    Serial.begin(9600);
    pinMode(kirmiziLed, OUTPUT);
    pinMode(yesilLed, OUTPUT);
    pinMode(maviLed, OUTPUT);
}
void loop()
{
    sensorSicaklik=analogRead(0);
    if(sensorSicaklik > 400 && sensorSicaklik < 520)
    {
        digitalWrite(yesilLed, LOW);
        digitalWrite(kirmiziLed, HIGH);
        digitalWrite(maviLed, HIGH);
        Serial.println("ölçülen sıcaklık oda sıcaklığında");
    }
    else if(sensorSicaklik > 0 && sensorSicaklik <= 400)
    {
        digitalWrite(yesilLed, HIGH);
        digitalWrite(kirmiziLed, LOW);
        digitalWrite(maviLed, HIGH);
        Serial.println("ölçülen sıcaklık oda sıcaklığından yüksek");
    }
    else if(sensorSicaklik>520 )
    {
        digitalWrite(yesilLed, HIGH);
        digitalWrite(kirmiziLed, HIGH);
        digitalWrite(maviLed, LOW);
        Serial.println("ölçülen sıcaklık oda sıcaklığından düşük");
    }
    delay(100);
}
```

Notlar:

.....

.....

.....

.....

.....

ÖĞRENCİNİN ADI		DEĞERLENDİRME				
ADI:	TEKNOLOJİ	İŞLEM BASAMAKLARI	İŞ ALIŞKANLIĞI	SÜRE	TOPLAM	
	%30	%30	%30	%10	Rakam ile	Yazı ile
	Ders Öğretmeni	Yunus ÖZEN	Koordinatör:	Yaşar SEVER		



Temrin Adı:

İleri – Geri Step Motor Kontrolü

Temrin No: 9

Devrede Kullanılan Malzemeler:

- 1x Arduino UNO
- 1x Step Motor
- 3x LED (KIRMIZI, SARI YEŞİL)
- 1X 470 Ω
- 3X 220 Ω

Açıklama:

Step Motor:

Elektrik enerjisini dönme hareketine çeviren elektromekanik bir cihazdır. Elektrik enerjisi alındığında rotor ve buna bağlı şaft, sabit açısal birimlerde (step-adım) dönmeye başlar. Step motorlar, sürücü ile çalışırlar ve bu sürücüler çok yüksek hızlı anahtarlama özelliğine sahiptir. Bu sürücü, bir encoder veya PLC'den giriş palsları alır. Alınan her giriş palsında, motor bir adım ilerler. Step motorları, bir motor turundaki adım sayısı ile adlandırılır. Örnek olarak 500 adımlık bir step motor, bir tam dönüşünde 500 adım yapar. Bu durumda bir adımın açısı $360/500 = 0.72$ derecedir. Bu değer, step motorun hassasiyetinin bir göstergesidir. Bir devirdeki adım sayısı yükseldikçe step motor hassasiyeti ve dolayısı ile maliyeti artar.

- Elde edilebilecek güç ve moment sınırlıdır.
- Adım açıları sabit olduğundan hareketleri sürekli değil darbelidir.
- Bir PLC ya da Encoder ile kontrol edilebilir.
- Açık döngülü olarak kontrol edilirler.
- Mekanik yapısı basit olduğundan bakım gerektirmezler.

Açıklama: Bu uygulamamızda arduinonun 2 numaralı pinine bağlı butona basıldığında step motor ileriye, 4 numaralı pinine bağlı buton basıldığında ise step motor geriye doğru dönmesi sağlanacaktır. Motor 8, 9, 10 ve 11 numaralı pinlere bağlanacaktır. Gerekli arduino yazılımı aşağıda verilmiştir.

ARDUİNO KODU:

```
int motorPin1 = 8; // 1. Bobinin bağlı olduğu pin
int motorPin2 = 9; // 2. Bobinin bağlı olduğu pin
int motorPin3 = 10; // 3. Bobinin bağlı olduğu pin
int motorPin4 = 11; // 4. Bobinin bağlı olduğu pin
int buton1=2; // 1. Butonun bağlı olduğu pin
int buton2=4; // 2. Butonun bağlı olduğu pin
void setup() {
  pinMode(motorPin1, OUTPUT); // Bobinleri çıkış olarak tanımlıyoruz.
  pinMode(motorPin2, OUTPUT); // Bobinleri çıkış olarak tanımlıyoruz.
  pinMode(motorPin3, OUTPUT); // Bobinleri çıkış olarak tanımlıyoruz.
  pinMode(motorPin4, OUTPUT); // Bobinleri çıkış olarak tanımlıyoruz.
  pinMode(buton1, INPUT); // Butonları giriş olarak tanımlıyoruz.
  pinMode(buton2, INPUT); // Butonları giriş olarak tanımlıyoruz.
}
void loop()
{
  if(digitalRead(buton2)==HIGH) // Buton 1'e basılı ise bu döngüyü uygula
  {
    digitalWrite(motorPin1, HIGH); // 1. Bobini 1 konumuna getiriyoruz.
    digitalWrite(motorPin2, LOW); // 2. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin3, LOW); // 3. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin4, LOW); // 4. Bobini 0 konumuna getiriyoruz.
    delay(5); // Bekleme süresi
    digitalWrite(motorPin1, LOW); // 1. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin2, HIGH); // 2. Bobini 1 konumuna getiriyoruz.
    digitalWrite(motorPin3, LOW); // 3. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin4, LOW); // 4. Bobini 0 konumuna getiriyoruz.
    delay(5); // Bekleme süresi
    digitalWrite(motorPin1, LOW); // 1. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin2, LOW); // 2. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin3, HIGH); // 3. Bobini 1 konumuna getiriyoruz.
    digitalWrite(motorPin4, LOW); // 4. Bobini 0 konumuna getiriyoruz.
    delay(5); // Bekleme süresi
    digitalWrite(motorPin1, LOW); // 1. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin2, LOW); // 2. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin3, LOW); // 3. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin4, HIGH); // 4. Bobini 1 konumuna getiriyoruz.
    delay(5); // Bekleme süresi
  }
  else{
    digitalWrite(motorPin1, LOW); // 1. Bobini 1 konumuna getiriyoruz.
    digitalWrite(motorPin2, LOW); // 2. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin3, LOW); // 3. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin4, LOW); // 4. Bobini 0 konumuna getiriyoruz.
  }

  if(digitalRead(buton1)==HIGH) // Buton 2'ye basılı ise bu döngüyü uygula
  {
    digitalWrite(motorPin1, LOW); // 1. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin2, LOW); // 2. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin3, LOW); // 3. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin4, HIGH); // 4. Bobini 1 konumuna getiriyoruz.
    delay(5); // Bekleme süresi
```

Notlar:

[illegible]

ÖĞRENCİNİN ADI	DEĞERLENDİRME					
ADI:	TEKNOLOJİ	İŞLEM BASAMAKLARI	İŞ ALIŞKANLIĞI	SÜRE	TOPLAM	
SOYADI:	%30	%30	%30	%10	Rakam ile	Yazı ile
	Ders Öğretmeni	Yunus ÖZEN	Koordinatör:	Yaşar SEVER		



Temrin Adı:

DA Motor Hız ve Yön Kontrolü

Temrin No: 10

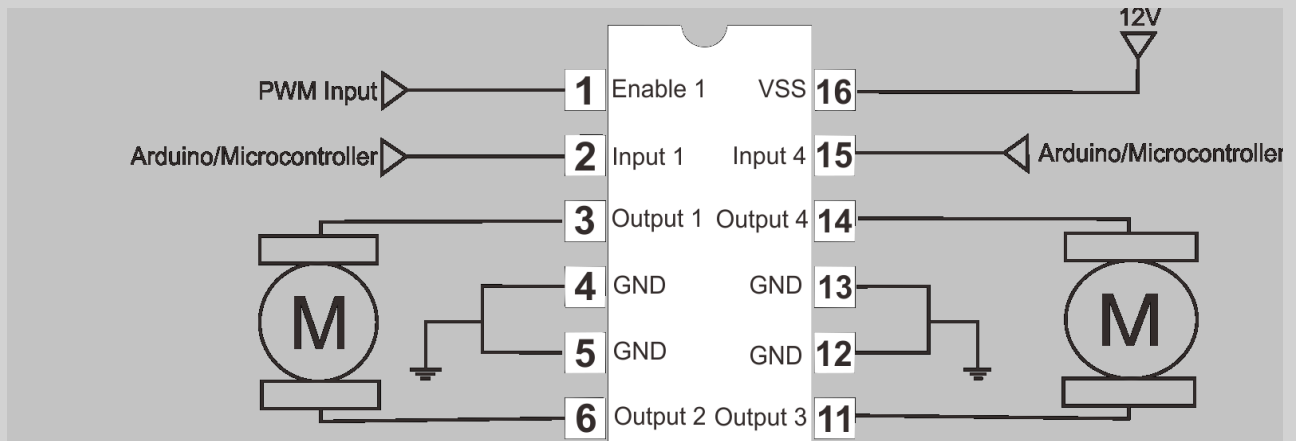
Devrede Kullanılan Malzemeler:

- 1x Arduino UNO
- 1x DA Motor
- 1x 1 k Ω
- 1x10 k Ω potansiyometre
- 1xL293D motor sürücü
- 1x buton

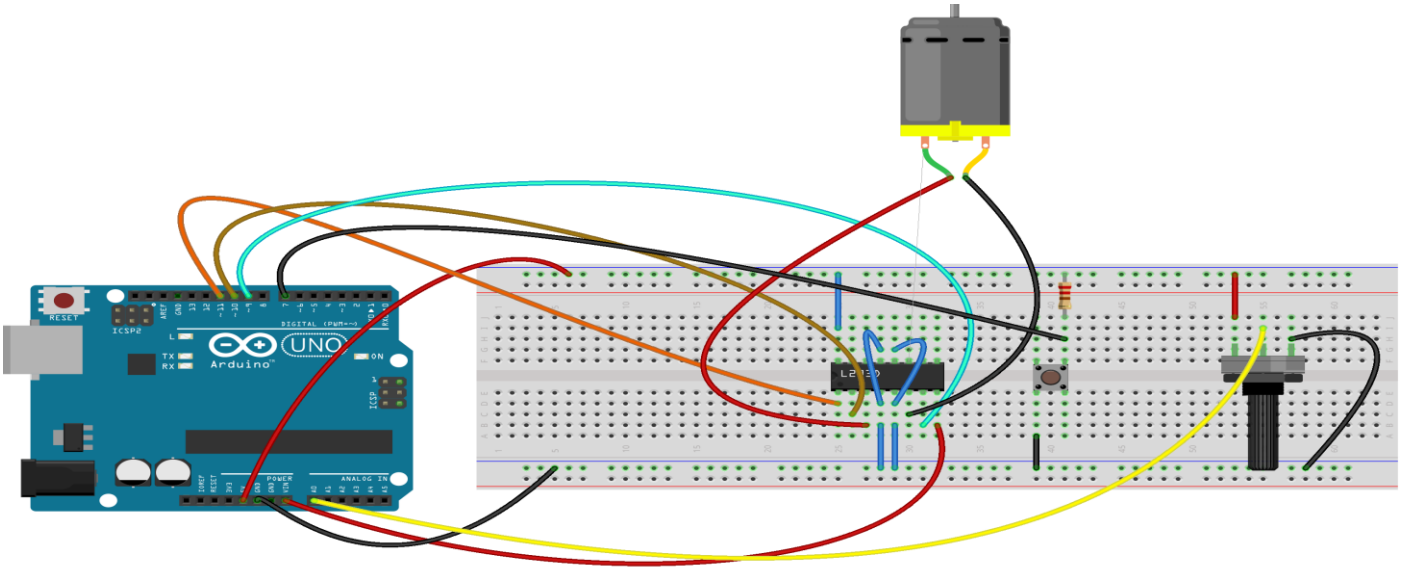
Açıklama: Arduino Uno nun 7 numaralı pinine bağlı olan butona basıldığında motor yönü değişecek, A0 pinine bağlı potansiyometreden ise istenilen hız ayarı yapılarak istenilen hızda ve yönde çalışması sağlanacaktır. Bu işlem için L293D entegresi kullanılacaktır.

Kısa Bilgi: L293D entegresini kullanarak Doğru Akım motorunu hızını ve devir yönünü kontrol edilmektedir. Bu entgre ile 2 ayrı DA motoru kontrol edilebilme olanağı sunmaktadır. Bu avantajından dolayı çizgi izleyen, bluetooth kontrollü, engelden kaçan, labirent çözen robotlar gibi çeşitli robotların motor sürücü kısmında yaygın olarak kullanılmaktadır. Bu entegrenin input 1 ve input 2 girişleri, motorun döneceği yönü; enable pini ise hangi çıkışların aktif olacağını kontrol ediyor. Enable pinine uygulayacağımız PWM sinyal, motorların hızını değiştirmemize olanak sağlıyor.

Pin yapısı aşağıdaki gibidir;



Devre Şeması:



Arduino Kodu:

```
int enablePin = 11;
int in1Pin = 10;
int in2Pin = 9;
int butonPin = 7;
int potPin = 0;

void setup()
{
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);
  pinMode(enablePin, OUTPUT);
  pinMode(butonPin, INPUT);
}

void loop()
{
  int hiz = analogRead(potPin) / 4;
  boolean ters = digitalRead(butonPin);
  motorCalistir(hiz, ters);
}

void motorCalistir(int hiz, boolean ters)
{
  analogWrite(enablePin, hiz);
  digitalWrite(in1Pin, ! ters);
  digitalWrite(in2Pin, ters);
}
```

[illegible]

ÖĞRENCİNİN ADI	DEĞERLENDİRME					
ADI:	TEKNOLOJİ	İŞLEM BASAMAKLARI	İŞ ALIŞKANLIĞI	SÜRE	TOPLAM	
SOYADI:	% 30	% 30	% 30	% 10	Rakam ile	Yazı ile
	Ders Öğretmeni	Yunus ÖZEN	Koordinatör:	Yaşar SEVER		



Devrede Kullanılan Malzemeler:

- 1x Arduino UNO
- 1x Servo Motor
- 1x10 kΩ potansiyometre

Kısa Bilgi: Öncelikle servo motoru tanımlamak gerekirse; Servo motorlar kullanıcının komutlarını yerine getiren motorlardır. Komutlar, pozisyon ve hız komutları veya hız ve pozisyonun birleşimi olabilir. 3 girişi bulunur. Bunlar güç, toprak ve sinyal girişleridir.

Arduino'nun **servo.h** kütüphanesini kullanarak servo motorları kontrol edebiliriz. Bu kütüphaneden kullandığımız bazı fonksiyonlar aşağıda açıklaması verilmiştir.

servomotor.attach(pin); fonksiyonunda servo motorun hangi pinden sinyal alacağını belirleriz.

servomotor.write(pos); fonksiyonunda servo motorun açısını ayarlamak için kullanılır.

Açıklama1: Arduino Uno nun 9 numaralı pinine bağlı servo motorun dönme açısı 180° ye ulaştığında 1 saniye bekleyecek ve geriye doğru dönmeye başlayacak 0° dereceye ulaştığında işe 5 saniye bekleyip tekrar tekrar hareketini sürdürecektir. Gerekli arduino kodunu yazarak programlama kartına yükleyiniz.

Arduino Kodu1:

```
#include <Servo.h>
```

```
Servo servomotor1;
```

```
int aci;
```

```
void setup()
```

```
{  
  servomotor1.attach(9);  
  servomotor1.write(0);  
}
```

```
void loop()
```

```
{  
  for (aci = 0; aci < 180; aci+=1) {  
    servomotor1.write(aci);  
    delay(10);  
  }
```

```
  for (aci = 180; aci > 0; aci-=1) {  
    servomotor1.write(aci);  
    delay(10);  
  }  
}
```

Açıklama2: Arduino Uno nun 9 numaralı pinine bağlı sevro motoru Analog0 pinine bağlı potansiyometre değerini değiştirerek motor dönmesini sağlayan arduino kodunu yazıp programlama kartına yükleyiniz.

Arduino Kodu2:

```
#include <Servo.h>

int potPin = A0;
int servoPin = 9;
Servo servo;

void setup() {
  pinMode(potPin, INPUT);
  servo.attach(servoPin);
}

void loop() {
  int okunan = analogRead(potPin);
  int aci = map(okunan, 0, 1023, 0, 180);
  servo.write(aci);
}
```

ÖĞRENCİNİN ADI	DEĞERLENDİRME					
ADI:	TEKNOLOJİ	İŞLEM BASAMAKLARI	İŞ ALIŞKANLIĞI	SÜRE	TOPLAM	
SOYADI:	%30	%30	%30	%10	Rakam ile	Yazı ile
	Ders Öğretmeni	Yunus ÖZEN	Koordinatör:	Yaşar SEVER		